Complexity Dichotomies for Counting Problems

Stylianos Despotakis, Andreas-Nikolas Göbel

$\mu\Pi\lambda\forall$

June 2012

Ladner's Theorem

Theorem (Ladner)

If $\mathrm{P} \neq \mathrm{NP}$, then there are problems in NP that are neither in P nor NP-complete.

Natural candidates: Graph Isomorphism, Integer Factorization

- Despite Ladner's theorem, most natural problems are either in P or NP-complete.
- Our goal is to find big classes of natural problems such that each problem in these classes is either tractable or very hard and it is easy to decide which of these cases holds.

Dichotomies for Decision Problems

- The H-Coloring Problem:
  - An H-coloring of G is just a homomorphism $G \to H$.
  - The H-coloring problem is in P if H is bipartite, otherwise it is NP-complete (Hell, Nešetřil '90).
- Constraint Satisfaction Problem
  - Given a set of constraints over subsets of n variables, find an assignment that satisfies all the constraints.
  - We have a dichotomy when the range of the variables is the Boolean Domain $\{0, 1\}$ (Shaeffer '78).
  - A dichotomy also holds when the range of the variables has three elements $\{0, 1, 2\}$ (Bulatov '03).
  - The general case (Feder-Vardi conjecture '93) remains yet unresolved.

Ladner's Theorem for Counting Problems

Ladner's theorem can be extended to counting problems as well:

Theorem (Ladner)

If $\mathrm{FP} \neq \#\mathrm{P}$, then there are problems in $\#\mathrm{P}$ that are neither in FP nor $\#\mathrm{P}$-complete.

Natural candidates: $\#\textsc{Div}$

- Like the decision case we are interested in finding broad classes of natural problems such that each problem in these classes is either tractable or very hard and it is easy to decide which of these cases holds.

Graph Homomorphism

Definition

Let $\mathbf{A} = (A_{i,j}) \in \mathbb{C}^{q \times q}$ be a complex matrix.
The graph homomorphism problem $\mathrm{EVAL}(\mathbf{A})$ is:
Input: A graph $G = (V, E)$.
Output: The partition function

$$Z_{\mathbf{A}}(G) = \sum_{\sigma: V \to [q]} \prod_{(u,v) \in E} A_{\sigma(u),\sigma(v)}.$$

If $\mathbf{A}$ is symmetric and G is an undirected graph, then we have the undirected graph homomorphism problem.

Examples

The function $Z_{\mathbf{A}}(G)$ can express many interesting properties.

- Let $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, then $Z_{\mathbf{A}}(G)$ counts the number of vertex covers in G.
- Let $\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$, then $Z_{\mathbf{A}}(G)$ counts the number of 3-colorings in G.

More Examples

- k-colorings: $\mathbf{A} = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix}$.
- Counting the number of induced subgraphs of G with an even number of edges: $\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

#CSP

Definition

Let $[q]$ be a domain set.
A constraint language F is a finite set of functions $\{f_1, \cdots, f_h\}$ in which $f_i : [q]^{r_i} \to \mathbb{C}$ is an r_i-ary function for some $r_i \geq 1$.
The $\#\mathrm{CSP}(\mathcal{F})$ is:
Input: Let $x = (x_1, \cdots, x_n)$ be a set of n variables. The input is a collection I of m tuples $(f, i_1, \cdots, i_r)$ in which f is an r-ary function in F and $i_1, \cdots, i_r \in [n]$.
Output:

$$Z_F(I) = \sum_{x \in [q]^n} \prod_{(f, i_1, \cdots, i_r) \in I} f(x_{i_1}, \cdots, x_{i_r}).$$

If the functions in F have as range the set $\{0, 1\}$, then we can see them as relations and we have the unweighted $\#\mathrm{CSP}(\mathcal{F})$.

Example

Let $[q] = \{0, 1\}$ and $F = \{f_0, f_1, f_2, f_3\}$, where

$$\begin{aligned} f_0(x,y,z) &= x \vee y \vee z, \\ f_1(x,y,z) &= \bar{x} \vee y \vee z, \\ f_2(x,y,z) &= \bar{x} \vee \bar{y} \vee z, \\ f_3(x,y,z) &= \bar{x} \vee \bar{y} \vee \bar{z}, \end{aligned}$$

then $\#\mathrm{CSP}(\mathcal{F})$ is the $\#3SAT$ problem.

Why is it a generalization of Graph Homomorphism?

- Let $\mathcal{F}$ contain a single binary function expressed by the matrix **A**.
- Also, let the variables $x_1, \cdots, x_n$ correspond to the vertices of the graph G and the constraints in I correspond to the edges of G.
- Then we can clearly see that Graph Homomorphism is a special case of $\#\mathrm{CSP}$.

Holant framework

Definition

Let $[q]$ be a domain set and F be a finite set of complex-valued functions over $[q]$.

The Holant(F) problem is:

Input: A signature grid $\Omega = (G, F, \pi)$, where $G = (V, E)$ is a labeled graph and π labels each vertex $v \in V$ with a function $f_v \in F$ so that the arity of f_v is the same as the degree of v.

Output:

$$Holant_\Omega = \sum_{\sigma : E \to [q]} \prod_{v \in V} f_v(\sigma \mid_{E(v)}).$$

Examples

- Perfect Matchings:
  - Let $[q] = \{0, 1\}$ and F contain all the $\textsc{ExactOne}$ functions.
  - At every vertex of G we attach the $\textsc{ExactOne}$ function with the appropriate arity.
  - The product $\prod_{v \in V} f_v(\sigma |_{E(v)})$ evaluates to 1 if $\sigma^{-1}(1) \subseteq E$ is a perfect matching and it evaluates to 0 otherwise.
  - Hence in this case, Holant_Ω counts the number of perfect matchings.
  - Perfect Matchings can't be expressed as Graph Homomorphism. (Freedman, Lovász, Schrijver '07)
- If we use the $\textsc{AtMostOne}$ function at every vertex, then we are counting all (not necessarily perfect) matchings.

Freely available variables

Definition

Let A be a set of functions. We can define a sub-framework $\mathrm{Holant}^{\mathcal{A}}$ of Holant as

$$\mathrm{Holant}^{\mathcal{A}}(\mathcal{F}) = \mathrm{Holant}(\mathcal{F} \cup \mathcal{A}).$$

In this sub-framework, we call the functions in $\mathcal{A}$, freely available functions.

If all equality functions are assumed to be freely available, then the sub-framework which is created is exactly the $\#\mathrm{CSP}$ problem, i.e.

$$\#\mathrm{CSP}(\mathcal{F}) = \mathrm{Holant}(\mathcal{F} \cup \mathit{Equalities}).$$

Why is it a generalization of $\#\mathrm{CSP}$?

- Represent an instance of $\#\mathrm{CSP}$ by a bipartite graph where the LHS is labeled by variables and the RHS is labeled by constraints (functions).
- The signature grid Ω is as follows: Every variable node on LHS is attached an $\textsc{Equality}$ function and every constraint node on RHS has the given constraint function.
- The $\textsc{Equality}$ function on each variable node forces the incident edges to take the same value; this effectively reduces edge assignments to vertex assignments assigning values to each variable on LHS as in $\#\mathrm{CSP}$.

Holant* and Holant^c

Definition

Let U denote the set of all unary functions. Then,

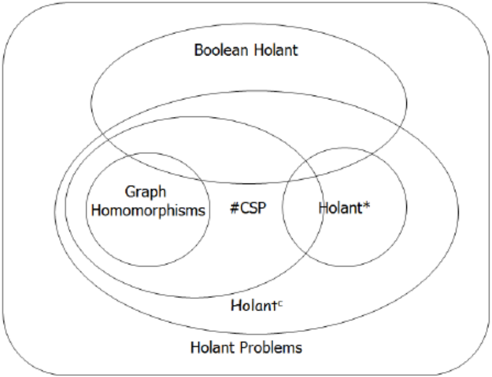
$$\mathrm{Holant}^\star(\mathcal{F}) = \mathrm{Holant}(\mathcal{F} \cup U).$$

Definition

Let Δ_i be the unary function which gives value 1 on inputs $i \in [q]$, and 0 on all other inputs. Then,

$$\mathrm{Holant}^c(F) = \mathrm{Holant}(\mathcal{F} \cup \{\Delta_1, \cdots, \Delta_q\}).$$

- Clearly, Holant^c is a super framework of $\mathrm{Holant}^\star$.
- Holant^c can be also viewed as a super framework of $\#\mathrm{CSP}$ (Dyer, Goldberg, and Jerrum '07).


Boolean Holant
Graph Homomorphisms
#CSP
Holant*
Holant^c
Holant Problems

Undirected (Symmetric) Graph Homomorphisms

Dyer, Greenhill '00

The counting Graph Homomorphisms problem of an undirected graph $H_{\mathbf{A}}$ is in P if every connected component of $H_{\mathbf{A}}$ is complete or complete bipartite. Otherwise, the problem is $\#\mathrm{P}$-complete.

- Bulatov and Grohe '05 extended the result to the non-negatively weighted setting.
- Goldberg, Grohe, Jerrum and Thurley provided a dichotomy theorem for all reall valued symmetric matrices, 73 pages long.

Cai, Chen, Lu '10

Let $\mathbf{A}$ be a symmetric complex matrix. Then counting graph Homomorphisms to $H_{\mathbf{A}}$ either can be computed in polynomial time or is $\#\mathrm{P}$-hard.

Directed Graph Homomorphisms

- Dyer, Goldberg and Paterson '07 proved a dichotomy for an acyclic family of graphs.
- Cai and Chen '10 solved the problem for all acyclic graphs with non-negative weighted.
- As discussed on previous slides, Graph Homomorphism is a special case of $\#\mathrm{CSP}$, therefore the remaining cases are included in the $\#\mathrm{CSP}$ dichotomiy theorems.

Boolean $\#\mathrm{CSP}$

- If the function is unweighted (relation) then the only tractable problems for $\#\mathrm{CSP}$ is the ones that contain only affine relations.
- For the weighted case we will need the following definitions:
    - We generalize the affine relations to pure affine functions by scaling them by a factor c.
    - Let $\mathcal{P}$ the the class of functions expressible as a product of unary functions, binary equality functions and binary disequality functions.

Dyer, Goldberg, Jerrum '07

Let $\mathcal{F}$ be a set of non-negative functions over Boolean domain. Then $\#\mathrm{CSP}(\mathcal{F})$ is $\#\mathrm{P}$-hard unless all the functions in $\mathcal{F}$ are pure affine or of the product type $\mathcal{P}$, in which case the problem is in P.

- Cai, Lu, Xia '09 and independently Bulatov, Dyer, Goldberg, Jalsenius and Richerby '09 finaly proved the dichotomy for complex weighted Boolean $\#\mathrm{CSP}$.

$\#\mathrm{CSP}$: The General Case

- For the unweighted setting Bulatov '08, with a proof using universal algebra, showed that $\#\mathrm{CSP}(\mathcal{F})$ is in FP, if $\mathcal{F}$ satisfies a citerion called congruence singularity, otherwise it is $\#\mathrm{P}$-complete.
- Dyer and Richerby '10 gave an alternative proof, w/o the use of universal algebra, providing an equivalent dichotomy criterion, and proving its decidability.
- Bulatov, Dyer, Goldberg, Jalsenius, Jerrum, and Richerby '10 showed that the above criterion can be extended to include positive rational weights.

$\#\mathrm{CSP}$: The General Case (concl.)

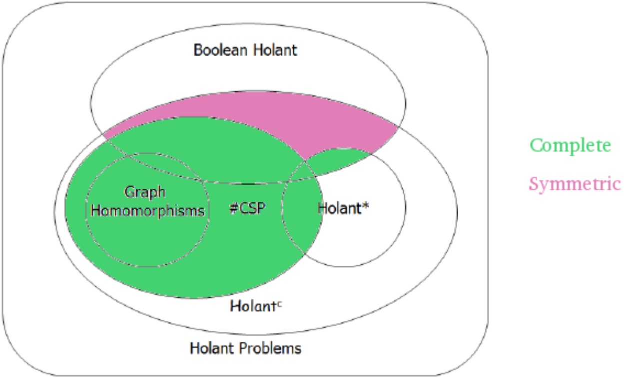
- The previous results were improved by Cai, Chen, Lu '11, where a dichotomy was provided for the non-negative weighted case.
- Finally the comlexity of $\#\mathrm{CSP}$ with complex weights was resolved by Cai and Chen '12:
  - The dichotomy criterion consists of 3 conditions. Any problem satisfying all 3 of them is in FP. Otherwise if either of them is violated then the problem is $\#P$ complete.
  - The decidability of the above 3 conditions remains open.

  Be sure not to miss Stelios's final talk on the details of the latter!

Dichotomies for Holant Problems

- Cai, Lu and Xia '09 gave a dichotomy for symmetric complex weighted boolean $\mathrm{Holant}^{\star}$;
  They completed the dichotomy for boolean (not only symmetric) $\mathrm{Holant}^{\star}$ in '11.
- In the same paper the gave a dichotomy for symmetric real-valued boolean Holant^{c};
  This was extended to the complex valued case by Cai, Huang and Lu '10

Summary


Boolean Holant
Graph Homomorphisms
#CSP
Holant*
Holant^c
Holant Problems
Complete
Symmetric

Perfect Matchings

Definition

The #PerfMatch problem is:
Input: A weighted undirected graph $G = (V, E, W)$.
Output:

$$\mathrm{PerfMatch}(G) = \sum_{E'} \prod_{(i,j)\in E'} w_{i,j},$$

where the summation is over all perfect matchings E' of G.

- If the weigths are all equal to 1, then $\mathrm{PerfMatch}(G)$ counts the number of perfect matchings in G.
- The decision version of the problem is in P, but the counting version is $\#\mathrm{P}$-complete (Valiant '79, Jerrum '87).

FKT algorithm

For planar graphs, $\#\textsc{PerfMatch}$ is in FP (Fisher, Kasteleyn, Temberley '61).

Theorem

There is a polynomial time computable function f that given a planar embedding of a planar graph $G = (V, E, W)$ defines $f : E \to \{-1, 1\}$ such that for the antisymmetric matrix M defined so that for all $i < j$

- *if $(i,j) \notin E$ then $M_{i,j} = M_{j,i} = 0$, and*
- *if $(i,j) \in E$ then $M_{i,j} = f(i,j) \cdot w_{i,j}$ and $M_{j,i} = -f(i,j) \cdot w_{i,j}$,*

it is the case that $\mathrm{PerfMatch}(G) = \sqrt{Det(M)}$.

Holographic reductions

Our strategy for a holographic reduction is the following:

- The individual components of an instance I of a counting problem (e.g. nodes and edges) will be replaced by gadgets that we call matchgates.
- Therefore, we transform the instance I to an instance Ω of what we call a matchgrid.
- The weigted sum of the perfect matchings of Ω will equal the number of solutions of I.

Matchgates

Definition

- A planar matchgate Γ is a triple (G, X, Y) where G is a planar embedding of a planar graph (V, E, W), where $X \subseteq V$ is a set of input nodes, $Y \subseteq V$ is a set of output nodes, and $X \cap Y = \emptyset$.
- The arity of the matchgate is $|X| + |Y|$.
- The standard signature of Γ with respect to a set $Z \subseteq X \cup Y$ is $\mathrm{PerfMatch}(G - Z)$.
- The standard signature of Γ is the $2^{|X|} \times 2^{|Y|}$ matrix $u(\Gamma)$ whose elements are the standard signatures of Γ with respect to Z for all possible choices of Z.

Bases

Definition

- A basis (of size 1) is a set of two distinct nonzero vectors, which we call n and p.
- The basis $\mathbf{b0} = [n, p] = [(0, 1), (1, 0)]$ is called the standard basis.
- In general, the vectors in a basis do not need to be independent.

From now on we will use as an example the basis $\mathbf{b1} = [n, p] = [(-1, 1), (1, 0)]$.

Tensor product

Definition

If we have two vectors q, r of length l, m, respectively, then we shall denote the tensor product $s = q \otimes r$ to be the vector s of length $l \cdot m$ in which $s_{i \cdot m+j} = q_i \cdot r_j$.

Thus, for example, for the basis $\mathbf{b1}$, $n \otimes p = (-1, 0, 1, 0)$.

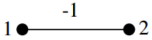
Generators and Recognizers

Definition

We say that a matchgate is a generator if it has zero input nodes and nonzero output nodes.
Similarly, we say that a matchgate is a recognizer if it has zero output nodes and nonzero input nodes.

Example of a generator

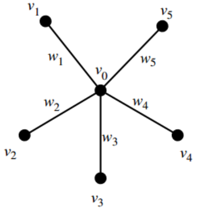
A generator mathgate Γ for basis **b1** with output nodes $\{1, 2\}$ and one edge of weight -1:



- The standard signature $u(\Gamma)$ is the vector (-1,0,0,1).
- Therefore, it generates $n \otimes n + n \otimes p + p \otimes n$.
- The signature of this generator with respect to the basis **b1** will then be (1,1,1,0).
- For $x \in \{n, p\}^2$ we shall denote by $valG(\Gamma, x)$ the signature element corresponding to x.
- Thus, for the current example, $valG(\Gamma, n \otimes p) = 1$ and $valG(\Gamma, p \otimes p) = 0$.

Example of a recognizer

A recognizer mathgate for basis **b1** with input nodes $v_1, v_2, \cdots, v_5$ and edge weights $w_1, w_2, \cdots, w_5$:



- Let us suppose a recognizer like the above but with k inputs.
- The purpose of such recognizers is to have *PerfMatch* take on appropriate values as the inputs range over 2^k possible tensor product values $x = x_1 \otimes x_2 \otimes \cdots \otimes x_k$, where $x_i \in \{n, p\}$.
- If vector u is the standard signature of Γ, then $valR(\Gamma, x)$ is the inner product of u and x.

Example of a recognizer (cont.)

Proposition

For all $k > 0$ and for all $w1, \cdots, w_k$ there exists a $k - input$ recognizer matchgate Γ such that on input $x = x_1 \otimes x_2 \otimes \cdots \otimes x_k \in \{n, p\}^k$ over basis **b1**, $valR(\Gamma, x)$ equals:

- $-(w_1 + \cdots + w_k)$ if $x_1 = \cdots = x_k = n$,
- w_i if $x_i = p$, and $x_j = n$ for every $j \neq i$,
- 0 otherwise.

Matchgrid

We define a matchgrid over a basis $\mathbf{b}$ to be a weighted undirected planar graph G that consists of:

- a set B of g generator matchgates $B_1, \cdots, B_g$,
- a set A of r recognizer matchgates $A_1, \cdots, A_r$, and
- a set C of f connecting edges $C_1, \cdots, C_f$ where each C_i edge has weight one and joins an output node in a generator with an input node of a recognizer.

Consider such a matchgrid $\Omega = (A, B, C)$ and denote by $X = \mathbf{b}^f = (n, p)^f$ the set of 2^f possible combinations of the basis elements n, p that can be transmitted simultaneously along the f connecting edges in the matchgrid.

The value of the matchgrid at x is the quantity

$$Holant(\Omega) = \sum_{x \in b^f} [\prod_{1 \le j \le g} valG(B_j, x_j)] \cdot [\prod_{1 \le i \le r} valG(A_i, x)].$$

Valiant's theorem

Theorem (Valiant '04)

For any matchgrid Ω over any basis $\mathbf{b}$ if Ω has weighted graph G then

$$\mathit{Holant}(\Omega) = \mathit{PerfMatch}(G).$$

#X-Matchings

Consider the problem #X-Matchings:

- Input: A Plannar Wighted Bipartite Graph $G = \{V = (V_1, V_2), E, W\}$, and the nodes in V_1 have degree 2.
- Output: $\sum_M \mathrm{mass}(M)$, where M is a (not necessarily perdect) matching and

$$\mathrm{mass}(M) = \prod_{(i,j)\in M} w_{i,j} \prod_{i \text{ unsat}} [-\sum_{j\sim i} w_{i,j}]$$

- Construct a matchgrid H over **b1** by replacing: $v \in V_1$ with the generator matchgate of the example and $u \in V_2$ with the recognizer matchgate of the example of proper degree.
- For each edge (u, v) connect an output of the generator matchgate of u to an input of the recognizer matchgate for v.
- $Holant(\Omega_H) = \#\text{X-Matchings}(G)$

Other Problems

- #PL-3-NAE-SAT: Plannar not-all-equal 3 satisfiability (CNF).
- $\#_7$PL-RTW-MON-3-CNF:
  - Input: A plannar 3CNF Boolean formula where each variable appears positively and in exactly two clauses (Plannar, Read-twice, monotone, 3CNF).
  - Output: Number of sat. assignments.

The suprising about the latter is that solving it mod 2 is $\oplus$P-complete!

Constructing a Holographic Algorithm

In order to design a Holographic Algorithm for a combinatorial problem, one has to do the following:

1. Find suitable basis vectors.
2. Find the suitable matchgates.

A series of results by Cai, Choudhary, Lu, Xia explore the realizable signatures, that is the functions for wich we can find a matchgate with the proper values under the relevant basis.

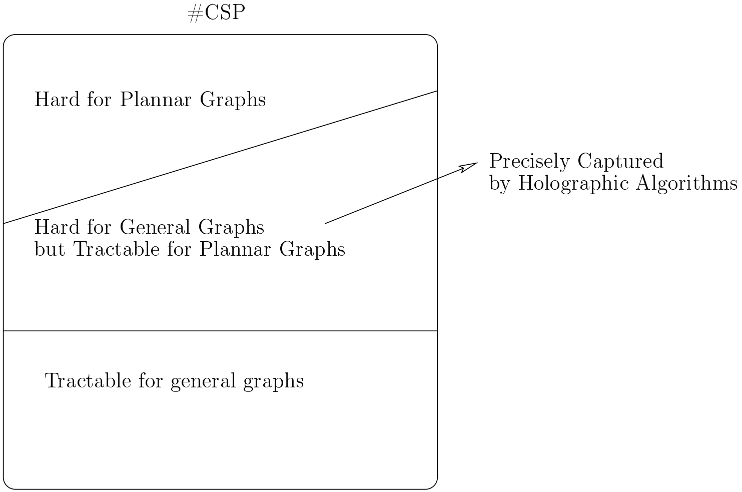
Not only plannar and relation to Holant_Ω

- A Holographic Algorithm uses plannar perfect mathcings on a plannar matchgrid in order to compute the output.
- This usually restricts the problem to plannar combinatorial instances.
- Cai, Lu, Xia '08 extended the Holographic algorithms by introducing a new gadget in order to realize signatures called fibonacci gate.
- Some new problems that can be solved in polynomial time are the following:
  - Given a 3-regular graph, compute the number of even colorings minus the number of odd colorings. (2-coloring on the edges, even when even edges are colored black).
  - Given a RTW-CNF formula, compute the number of even lying assignments minus the number of odd lying assignments. (A variable is lying when it is incosistent).

Relation to Holant problems and the Ising Model

- The $\mathrm{Holant}_{\Omega'}$ $(\Omega' = (G, F, \pi))$ problem defined in a previous section is actually the Holant_{Ω} function defined by Valant.
  - Every vertex v of $\mathrm{Holant}_{\Omega'}$ problem can be replaced by a matchgate, with value equal to f_v.
  - The degree of v equals to the number of output nodes of the matchgate.
  - Assignments to edges of G corresponds to vectors of the basis used by the matchgrid.
- The Ising Model from statistical Physics Can be formulated as a Holant problem
  - For planar graphs, the Ising model is exactly solvable by a holographic reduction to the FKT algorithm.

Computational Power



To Be Continued

- In Our next and hopefully final talks we will discuss the following:
  - Stelios: Technical Details for some of the forementioned Dichotomies. July 20th ?
  - Andreas: Approximating $\#\mathrm{CSP}$. July 23rd ?
- We encourage everyone to be there!

THANK YOU!!

Till Next Time!